

JAPAN PATENT OFFICE

This is to certify that the annexed is a true copy of the following application as filed with this Office.

Date of Application: April 1,2002

Application Number: Japanese Patent Application
No.2002-099102

Applicant(s): NTT DoCoMo, Inc.

Commissioner,
Patent Office

(Seal)

(Document Name) Patent Application
(Reference Number) 13-0732
(Presentation Date) April 1,2002
(Directly) Commissioner of the Patent Office
(IPC) G06F 9/06

(Inventor)

(Residence or Address)

c/o NTT DoCoMo, Inc., Nagatacho 2-11-1,
Chiyoda-ku, Tokyo

(Name) Koichi MORIYAMA

(Inventor)

(Residence or Address)

c/o NTT DoCoMo, Inc., Nagatacho 2-11-1,
Chiyoda-ku, Tokyo

(Name) Takashi KADOHIRO

(Inventor)

(Residence or Address)

c/o NTT DoCoMo, Inc., Nagatacho 2-11-1,
Chiyoda-ku, Tokyo

(Name) Tadao TAKAMI

(Inventor)

(Residence or Address)

c/o NTT DoCoMo, Inc., Nagatacho 2-11-1,
Chiyoda-ku, Tokyo

(Name) Seiji HOSHI

(Applicant)

(Identification Number) 392026693

(Name) NTT DoCoMo, Inc.

(Attorney)

(Identification Number) 100088155

(Patent Attorney)

(Name) Yoshiki HASEGAWA

(Attorney)

(Identification Number) 100092657

(Patent Attorney)

(Name) Shiro TERASAKI

(Attorney)

(Identification Number) 100114270

(Patent Attorney)

(Name) Tomoya KUROKAWA

(Attorney)

(Identification Number) 100108213

(Patent Attorney)

(Name) Toyotaka ABE

(Attorney)

(Identification Number) 100113549

(Patent Attorney)

(Name) Mamoru SUZUKI

(Official Fee)

(Pre-Paid Master Note Number) 014708

(Amount to be paid) 21000

(Lists of the Article to be presented)

(Name of Article)	Specification	1
-------------------	---------------	---

(Name of Article)	Drawing	1
-------------------	---------	---

(Name of Article)	Abstract	1
-------------------	----------	---

(Proof Reading) Required

[Document Name] Specification

[Title of the Invention] SOFTWARE UPDATE METHOD FOR
OF COMMUNICATION TERMINAL, COMMUNICATION
TERMINAL, AND SOFTWARE UPDATE SYSTEM

5 [Claims]

[Claim 1] An update method for software which is
stored in a rewritable non-volatile memory of a
communication terminal, provided with said rewritable
non-volatile memory and a rewritable volatile memory, and is
10 directly executed, comprising:

an update file transfer step of transferring an update software
from a software management server for managing the update
software to be stored in the rewritable non-volatile memory of
said communication terminal to said communication terminal
15 as an update file, and storing the update file in the rewritable
volatile memory of said communication terminal; and

a software rewriting step of rewriting the software, which is
stored in said rewritable non-volatile memory and is directly
executed, with the update software which is stored in said
20 rewritable volatile memory as the update file.

[Claim 2] The update method for software according to
Claim 1, further comprising:

a transfer status transition step of limiting operation of
software which uses said rewritable volatile memory before
25 transferring the update file in said update file transfer step,
and securing an area required for storing said update file in

said rewritable volatile memory; and
a transfer request step of said communication terminal
requesting transfer of the update file to said software
management server after the area required for storing said
5 update file is secured in said rewritable volatile memory,
wherein said update file transfer step is started based on the
transfer request of the update file by said communication
terminal.

[Claim 3] The update method for software according to
10 Claim 2, further comprising:

an update file information transmission step of said software
management server transmitting update file information
including the size information of said update file; and

an update file information reception step of said
15 communication terminal receiving said update file
information,

wherein in said transfer status transition step, operation of the
software which uses said rewritable volatile memory is
limited based on the size information included in said update
20 file information, and an area where said update file is stored
is secured in said rewritable volatile memory.

[Claim 4] The update method for software according to
one of Claim 1 to Claim 3, further comprising:

a communication terminal identification information

25 transmission step of said communication terminal transmitting
the identification information of said communication terminal

to a software management server;
a communication terminal identification information reception
step of said software management server receiving the
identification information of said communication terminal
5 which is transmitted by said communication terminal; and
an update software selection step of said software management
server selecting an update software to be stored in said
rewritable non-volatile memory of said communication
terminal, out of the update software owned by said software
10 management server based on the identification information of
said communication terminal which is transmitted by said
communication terminal,
wherein said software management server transmits the
selected update software to said communication terminal as an
15 update file.

[Claim 5] The update method for software according to one of Claim 1 to Claim 4, further comprising:

5 a software identification information transmission step of said communication terminal transmitting the identification information of software, which is stored in said rewritable non-volatile memory and is directly executed, to the software management server;

10 a software identification information reception step of said software management server receiving the identification information of the software transmitted by said communication terminal; and

15 an update software selection step of said software management server selecting an update software to be stored in said rewritable non-volatile memory of said communication terminal, out of the update software owned by said software management server based on the identification information of said software transmitted by said communication terminal,

20 wherein said software management server transmits the selected update software to said communication terminal as an update file.

[Claim 6] The update method for software according to Claim 5, further comprising a differential file creation step of said software management server creating a differential file between the software which is stored in said rewritable non-volatile memory of said communication terminal and is directly executed and the update software owned by said

25

software management server based on the identification information of the software transmitted by said communication terminal,

5 wherein said software management server transmits said differential file to said communication terminal as said update file.

10 [Claim 7] The update method for software according to one of Claim 1 to Claim 6, wherein the update file is transferred by radio communication in said update file transfer step.

15 [Claim 8] The update method for software according to one of Claim 1 to Claim 7, further comprising a restore status transition step of performing wire communication with the software restore device for restoring software, which is stored in said rewritable non-volatile memory and is directly executed, by transferring the software to said rewritable non-volatile memory, when the processing of rewriting the software, which is stored in said rewritable non-volatile memory and is directly executed, with the update software stored in said rewritable
20 volatile memory as an update file, failed in said software rewriting step.

 [Claim 9] A communication terminal comprising a rewritable non-volatile memory and a rewritable volatile memory, further comprising:

25 update file reception means for receiving an update software from a software management server, which manages

said update software, as an update file, and storing said update file in said rewritable volatile memory; and

software rewriting means for rewriting the software, which is stored in said rewritable non-volatile memory and is directly executed, with the update software, which is stored in said rewritable volatile memory as an update file, after storage of said update file to said rewritable volatile memory by said update file reception means completes.

[Claim 10] The communication terminal according to Claim 9, further comprising:

transfer status transition means for limiting operation of the software which uses said rewritable volatile memory and securing an area required for storing said update file in said rewritable volatile memory before said update file reception means receives the update file; and

transfer request transmission means for requesting transfer of an update file to said software management server after said transfer status transition means secures the area required for storing said update file in said rewritable volatile memory.

[Claim 11] The communication terminal according to Claim 10, further comprising update file information reception means for receiving the update file information including the size information of said update file from said software management server, wherein said transfer status transition means limits operation of the software which uses said

rewritable volatile memory based on the size information included in said update file information, and secures the area for storing said update file in said rewritable volatile memory.

5 [Claim 12] The communication terminal according to one of Claim 9 to Claim 11, further comprising:

communication terminal identification information storage means for storing identification information of said communication terminal; and communication terminal identification information transmission means for transmitting
10 identification information of said communication terminal which is stored in said communication terminal identification information storage means to the software management server.

[Claim 13] The communication terminal according to one of Claim 9 to Claim 12, further comprising:

15 software identification information storage means for storing the identification information of software which is stored in said rewritable non-volatile memory; and

software identification information transmission means for transmitting the identification information of said software
20 which is stored by said software identification information storage means to said software management server.

[Claim 14] The communication terminal according to one of Claim 9 to Claim 13, wherein said update file reception means receives said update file by radio communication.

25 [Claim 15] The communication terminal according to one of Claim 9 to Claim 14, further comprising:

rewrite success judgment means for judging whether the rewriting software executed by said software rewriting means succeeded;

5 restore status transition means for performing wire communication with a software restore device for restoring software, which is stored in said rewritable non-volatile memory and is directly executed, by transferring software to said rewritable non-volatile memory when said rewrite success judgment means judged that rewrite of the software executed
10 by said software rewrite means failed; and

 restore software reception means for receiving the software from said software restore device and storing said software in said rewritable non-volatile memory.

15 [Claim 16] A software update system, comprising the communication terminal according to one of Claim 9 to Claim 14, and a software management server for managing an update software to be stored in a rewritable non-volatile memory of said communication terminal.

20 [Claim 17] A software management system, comprising the communication terminal according to Claim 15, a software management server for managing an update software to be stored in a rewritable non-volatile memory of said communication terminal, and a software restore device for restoring the software in said rewritable non-volatile memory
25 by transferring the software to the rewritable non-volatile memory of said communication terminal by connecting with

said communication terminal via wire communication.

[Detailed Description of the Invention]

[0001]

[Technical Field to which the Invention Pertains]

5 The present invention relates to a software update method for updating the software of a communication terminal, a communication terminal and a software update system.

[0002]

10 [Prior Art]

 The Operating System (OS) for controlling a communication terminal and such basic software as middleware are normally stored in a non-volatile memory installed on the communication terminal, and is directly
15 executed on the non-volatile memory so as not to be lost when power is turned OFF, in order that operation is possible by turning the power ON again even if power of the communication terminal is shut OFF.

[0003]

20 For non-volatile memory, normally a rewritable non-volatile memory, such as a flash ROM (Read Only Memory) is used to avoid an exchange of the non-volatile memory itself when the basic software is upgraded, and the version upgrade is executed by rewriting the software.

25 [0004]

 Along with the recent advancements in communication

means, a method of upgrading software, which is stored in the above mentioned rewritable non-volatile memory and is directly executed, by downloading the update software, from a software management server which manages update software has been proposed. In this software update method, a method of rewriting the software, which is stored in the rewritable non-volatile memory and is directly executed, is used at the same time with downloading the update software. Therefore, if rewriting software stops mid-way by an interruption in the download of the update software due to such a cause as a power failure, it may become impossible to use the communication terminal. To prevent this risk, a conventional communication terminal has dual rewritable non-volatile memories. And to update the software, the currently operated software remains in one of the dual rewritable non-volatile memories, and in this status the update software is transferred to the other rewritable non-volatile memory from the software management server. Then after transfer of the update software completes, operation is switched from the rewritable non-volatile memory, where currently operated software is stored, to the rewritable non-volatile memory where the update software is stored. Therefore even if an update of software fails, as mentioned above, the currently operated software operates, so the risk of the communication terminal becoming disabled can be prevented.

[0005]

[Problem to be Solved by the Invention]

However in the conventional software update method,
the rewritable non-volatile memory of the communication
5 terminal must be duplicated to prevent risk due to the failure
of software update, as mentioned above. As a result, the
communication terminal size become large and cost increases.
With the foregoing in view, it is an object of the present
invention to provide a software update method, a
10 communication terminal and a software update system for
decreasing the risk of failing to update software and for
solving the above mentioned problems of size increase and
cost increase of the communication terminal.

[0006]

15 [Means for Solving the Problem]

To solve the above problems, the software update
method of the present invention is an update method for
software, which is stored in a rewritable non-volatile memory
of a communication terminal comprising the rewritable
20 non-volatile memory and a rewritable volatile memory and is
directly executed, comprising: an update file transfer step of
transferring an update software from a software management
server, for managing the update software to be stored in the
rewritable non-volatile memory of the communication
25 terminal, to the communication terminal as an update file, and
storing the update file in the rewritable volatile memory of

the communication terminal; and a software rewriting step of rewriting software, which is stored in the rewritable non-volatile memory and is directly executed, with the update software, which is stored in the rewritable volatile memory as the update file.

[0007]

According to the software update method of the present invention, the update file is transferred to the rewritable volatile memory of the communication terminal, and after transfer of the update file completes, the software, which is stored in the rewritable non-volatile memory and is directly executed, is rewritten with the update software, which is stored in the rewritable volatile memory as the update file. Therefore the rewritable non-volatile memory need not be duplicated for updating the software. As a result, the communication terminal can be downsized and cost can be decreased. Also transfer of an update file tends to take time and to have such risk factors as power disconnection, and rewriting of software which takes a shorter time than above can be separated. As a consequence, time for rewriting software can be decreased, and the risk of failing to update software can be decreased. Also even if transfer is interrupted in the middle of transferring the update file, the software, which is stored in the rewritable non-volatile memory of the communication terminal and is directly executed, has not been rewritten, so operation of the

communication terminal is not affected.

[0008]

The software update method of the present invention may further comprise a transfer status transition step of
5 limiting the operation of the software which uses the rewritable volatile memory before transferring the update file in the update file transfer step, and then securing an area required for storing the update file in the rewritable volatile memory, and a transfer request step of the communication
10 terminal requesting transfer of the update file to the software management server after the area required for storing the update file is secured in the rewritable volatile memory, wherein the update file transfer step is started based on the transfer request of the update file by the communication
15 terminal.

[0009]

According to the software update method of the present invention, operation of the software which uses the rewritable volatile memory is limited during the normal operating status
20 of the communication terminal before an update file is transferred, then the update file is transferred to the rewritable volatile memory. Therefore even if the capacity of the rewritable volatile memory is small, the update file can be transferred to the rewritable volatile memory.

[0010]

The software update method of the present invention

may further comprise an update file information transmission step of the software management server transmitting update file information including the size information of the update file, and an update file information reception step of the communication terminal receiving the update file information, wherein in the transfer status transition step, operation of the software which uses the rewritable volatile memory is limited based on the size information included in the update file information, and an area where the update file is stored is secured in the rewritable volatile memory.

[0011]

According to the software update method of the present invention, the area for storing the update file is secured in the rewritable volatile memory based on the size information of the update file included in the update file information. Therefore an appropriate area size for storing the update file can be secured in the rewritable volatile memory.

[0012]

The software update method of the present invention may further comprise a communication terminal identification information transmission step of the communication terminal transmitting the identification information of the communication terminal to a software management server, a communication terminal identification information reception step of the software management server receiving the identification information of the communication terminal

which is transmitted by the communication terminal, and an update software selection step of the software management server selecting an update software to be stored in the rewritable non-volatile memory of the communication terminal, out of the update software owned by the software management server based on the identification information of the communication terminal which is transmitted by the communication terminal, wherein the software management server transmits the selected update software to the communication terminals as an update file.

[0013]

According to the software update method of the present invention, the communication terminal transmits the identification information of the communication terminal to the software management server. Therefore the software management server can specify the model of the communication terminal. As a result, out of the update software owned by the software management server, update software to be stored in the rewritable non-volatile memory of the communication terminal can be accurately selected.

[0014]

The software update method of the present invention may further comprise a software identification information transmission step of the communication terminal transmitting the identification information of software, which is stored in the rewritable non-volatile memory and is directly executed,

to the software management server, a software identification
information reception step of the software management server
receiving the identification information of the software
transmitted by the communication terminal, and an update
5 software selection step of the software management server
selecting an update software to be stored in the rewritable
non-volatile memory of the communication terminal out of
the update software owned by the software management
server based on the identification information of the software
10 transmitted by the communication terminal, wherein the
software management server transmits the selected update
software to the communication terminal as an update file.

[0015]

15 According to the software update method of the present
invention, the communication terminal transmits the
identification information of software, which is stored in the
rewritable non-volatile memory and is directly executed, to
the software management server. Therefore the software
20 management server can specify the version of the software
which is currently operated in the rewritable non-volatile
memory of the communication terminal based on the
identification information of the software. As a result, out
of the update software owned by the software management
25 server, update software to be stored in the rewritable
non-volatile memory of the communication terminal can be

accurately selected.

[0016]

The software update method of the present invention may further comprise a differential file creation step of the software management server creating a differential file
5 between the software, which is stored in the rewritable non-volatile memory of the communication terminal and is directly executed, and the update software owned by the software management server based on the identification
10 information of the software transmitted by the communication terminal, wherein the software management server transmits the differential file to the communication terminal as an update file.

[0017]

15 According to the software update method of the present invention, the software management server creates a differential file between the software, which is stored in the rewritable non-volatile memory of the communication terminal, and the update software owned by the software
20 management server, based on the software identification information. Therefore by using the differential file as the update file, the size of the update file to be transmitted can be decreased. As a result, the update file can be accurately transmitted even via a narrowband communication line.

25 [0018]

In the software update method of the present invention,

the update file may be transferred by radio communication in the software transfer step. Since the software update method of the present invention has the above feature, the update file can be efficiently transferred even via a narrowband communication line, such as a radio communication line. Also even if it takes time to transfer an update file, this does not become a cause of failing to update software, therefore the risk of failing to update software is low. As a consequence, the rewritable non-volatile memory of the communication terminal need not be duplicated for updating software.

[0019]

The software update method of the present invention may further comprise a restore status transition step of performing wire communication with the software restore device for restoring the software, which is stored in the rewritable non-volatile memory and is directly executed, by transferring the software to the rewritable non-volatile memory, when the processing of rewriting the software, which is stored in the rewritable non-volatile memory and is directly executed, with the update software, which is stored in the rewritable volatile memory as an update file, failed in the software rewriting step.

[0020]

According to the software update method of the present invention, even if rewriting of software failed due to a power

failure, for example, which occurs in the middle of the processing of rewriting the software, which is stored in the rewritable non-volatile memory and is directly executed, with the update software, wire communication with the software restore device for transferring the software to the rewritable non-volatile memory and for restoring the software, which is stored in the rewritable non-volatile memory and is directly executed, becomes possible. Therefore if the communication terminal and the software restore device are connected, software can be restored in the rewritable non-volatile memory.

[0021]

The communication terminal of the present invention is a communication terminal comprising a rewritable non-volatile memory and a rewritable volatile memory, further comprising update file reception means for receiving an update software from a software management server, which manages the update software, as an update file, and storing the update file to the rewritable volatile memory, and software rewriting means for rewriting the software, which is stored in the rewritable non-volatile memory and is directly executed, with the update software, which is stored in the rewritable volatile memory as an update file, after storage of the update file to the rewritable volatile memory by the update file reception means completes.

[0022]

According to the communication terminal of the present invention, the update file is transferred to the rewritable volatile memory of the communication terminal, and after transfer of the update file completes, the software, which is stored in the rewritable non-volatile memory and is directly executed, is rewritten with the update software, which is stored in the rewritable volatile memory as an update file. Therefore the rewritable non-volatile memory need not be duplicated for updating the software. As a result, a communication terminal can be downsized and cost can be decreased. Also transfer of the update file, which takes time and tends to have such risk factors as power disconnection, and the rewriting of software which takes shorter time than above, can be separated. As a consequence, time for rewriting software can be decreased and the risk of failing to update software can be decreased. Also even if transfer is interrupted in the middle of transferring an update file, the software, which is stored in the rewritable non-volatile memory of the communication terminal and is directly executed, has not been rewritten, so operation of the communication terminal is not affected.

[0023]

The communication terminal of the present invention may further comprise transfer status transition means for limiting operation of the software which uses the rewritable volatile memory and secures an area required for storing the

update file in the rewritable volatile memory before the
update file reception means receives the update file, and
transfer request transmission means for requesting transfer of
an update file to the software management server after the
5 transfer status transition means secures the area required for
storing the update file in the rewritable volatile memory.

[0024]

According to the communication terminal of the
present invention, operation of the software which uses the
10 rewritable volatile memory is limited in the normal operating
status of the communication terminal before an update file is
transferred, then the update file is transferred to the
rewritable volatile memory. Therefore even if the capacity
of the rewritable volatile memory is small, the update file can
15 be transferred to the rewritable volatile memory.

[0025]

The communication terminal of the present invention
may further comprise update file information reception means
for receiving update file information including the size
20 information of the update file from the software management
server, wherein the transfer status transition means limits
operation of software which uses the rewritable volatile
memory based on the size information included in the update
file information, and secures the area for storing the update
25 file in the rewritable volatile memory.

[0026]

According to the communication terminal of the present invention, the area for storing the update file is secured in the rewritable volatile memory based on the size information of the update file included in the update software information. Therefore an appropriate size of the area for storing the update file can be secured in the rewritable volatile memory.

[0027]

The communication terminal of the present invention may further comprise communication terminal identification information storage means for storing identification of the communication terminal, and communication terminal identification information transmission means for transmitting identification information of the communication terminal, which is stored in the communication terminal identification information storage means, to the software management server.

[0028]

According to the communication terminal of the present invention, the communication terminal transmits the identification information of the communication terminal to the software management server. Therefore the software management server can specify the model of the communication terminal. As a result, out of the update software owned by the software management server, update software to be stored in the rewritable non-volatile memory of

the communication terminal can be accurately selected.

[0029]

The communication terminal of the present invention may further comprise software identification storage means for storing the identification information of software which is stored in the rewritable non-volatile memory, and software identification information transmission means for transmitting the identification information of the software which is stored by the software identification information storage means to the software management server.

[0030]

According to the communication terminal of the present invention, the communication terminal transmits the identification information of the software, which is stored in the rewritable non-volatile memory and is directly executed, to the software management server. Therefore the software management server which received the identification information of the software can specify the version of the software which is currently operated in the rewritable non-volatile memory of the communication terminal based on the identification information of the software. As a result, out of the update software owned by the software management server, update software to be stored in the rewritable non-volatile memory of the communication terminal can be accurately selected. It is also possible that the software management server creates a differential file

between the software, which is stored in the rewritable non-volatile memory of the communication terminal, and the update software owned by the software management server, based on the identification information of the software. If
5 the differential file is used as the update file, the size of the update file to be transmitted can be decreased. As a result, the update file can be accurately transmitted even via a narrowband communication line.

[0031]

10 In the communication terminal of the present invention, the update file reception means may receive the update file by radio communication. Since the communication terminal of the present invention has the above feature, the update file can be efficiently transferred even via a narrowband
15 communication line, such as a radio communication line. Also even if it takes time to transfer the update file, this does not become a cause of failing to update software, and thus, risk of failing to update software is low. As a consequence, the rewritable non-volatile memory of the communication
20 terminal need not be duplicated for updating software.

[0032]

The communication terminal of the present invention may further comprise rewrite success judgment means for judging whether rewriting software, executed by the software
25 rewriting means, succeeded, restore status transition means for performing wire communication with a software restore

device for restoring software, which is stored in the rewritable non-volatile memory and is directly executed, by transferring software to the rewritable non-volatile memory when the rewrite success judgment means judged that rewrite
5 of the software executed by the software rewrite means failed, and restore software reception means for receiving the software from the software restore device and storing the software to the rewritable non-volatile memory.

[0033]

10 According to the communication terminal of the present invention, the rewrite success judgment means judges whether the processing of rewriting the software, which is stored in the rewritable non-volatile memory and is directly executed, with the update software succeeded. When it was
15 judged that rewriting of the software failed as a result of judgment by the rewrite success judgment means, the restore status transition means enables wire communication with the software restore device for transferring the software to the rewritable non-volatile memory and for restoring software,
20 which is stored in the rewritable non-volatile memory and is directly executed. Therefore if the communication terminal and the software restore device are connected, the restore software reception section receives software from the software restore device and stores the software in the
25 rewritable non-volatile memory, so software for which rewriting failed can be restored.

[0034]

A software update system of the present invention comprises the communication terminal of the present invention and a software management server for managing an
5 update software to be stored in a rewritable non-volatile memory of the communication terminal.

[0035]

According to the software update system of the present invention, the update file is transferred to the rewritable
10 non-volatile memory of the communication terminal, and after transfer of the update file completes, the software, which is stored in the rewritable non-volatile memory and is directly executed, is rewritten with the update software, which is stored in the rewritable volatile memory, as an update file.
15 Therefore the rewritable non-volatile memory need not be duplicated for updating the software. As a result, the communication terminal can be downsized and cost can be decreased. Also transfer of the update file, which takes time and tends to have such risk factors as power disconnection,
20 and rewriting of software which takes shorter time than above, can be separated. As a consequence, time for rewriting software can be decreased, and risk of failing to update software can be decreased. Also even if a transfer is interrupted in the middle of transferring an update file, the
25 software, which is stored in the rewritable non-volatile memory of the communication terminal and is directly

executed, has not been rewritten, so operation of the communication terminal is not affected. Also operation of the software which uses the rewritable volatile memory is limited in the normal operating status of the communication terminal before the update file is transferred, then the update file is transferred to the rewritable volatile memory. Therefore even if the capacity of the rewritable volatile memory is small, the update file can be transferred to the rewritable volatile memory. Also the area for storing the update file is secured in the rewritable volatile memory based on the size information of the update file included in the update information. Therefore an appropriate size of an area for storing the update file can be secured in the rewritable volatile memory. Also the communication terminal transmits the identification information of the communication terminal to the software management terminal. Therefore the software management server can specify the model of the communication terminal. As a result, out of the update software owned by the software management server, update software to be stored in the rewritable non-volatile memory of the communication terminal can be accurately selected. Also the communication terminal transmits the identification information of software, which is stored in the rewritable non-volatile memory and is directly executed, to the software management server, and the software management server receives the identification information. Therefore the

software management server can specify the version of the software which is currently operated in the rewritable non-volatile memory of the communication terminal based on the identification information of the received software. As a result, out of the update software owned by the software management server, update software to be stored in the rewritable non-volatile memory of the communication terminal can be accurately selected. It is also possible that the software management server creates a differential file between the software, which is stored in the rewritable non-volatile memory of the communication terminal and is directly executed, and the update software which is owned by the software management server, based on the identification information of the software. If the differential file is used as the update file, the size of the update file to be transmitted can be decreased. As a result, the update file can be accurately transmitted even via a narrowband communication line. Also since the communication terminal of the present invention has the above feature, the update file can be efficiently transferred even via a narrowband communication line, such as a radio communication line. Also even if it takes time to transfer the update file, this does not become a cause of failing to update software, and risk of failing to update software is low. As a consequence, the rewritable non-volatile memory of the communication terminal need not be duplicated for updating software.

[0036]

The software update system of the present invention may be comprised of the communication terminal of the present invention, a software management server for managing the update software to be stored in a rewritable non-volatile memory of the communication terminal, and a software restore device for restoring the software in the rewritable non-volatile memory by transferring the software to the rewritable non-volatile memory of the communication terminal by connecting with the communication terminal via a wire communication.

[0037]

According to the software update system of the present invention, even if rewriting of software failed due to power failure, for example, which occurred in the middle of the processing of rewriting the software, which is stored in the rewritable non-volatile memory and is directly executed, with the update software, wire communication with the software restore device for transferring the software to the rewritable non-volatile memory and for restoring the software, which is stored in the rewritable non-volatile memory and is directly executed, is enabled. Therefore if the communication terminal and the software restore device are connected, software can be restored in the rewritable non-volatile memory.

[0038]

[Embodiments of the Invention]

The software update system according to embodiments of the present invention will now be described with reference to the accompanying drawings. Fig. 1 is a system block diagram depicting the physical composing elements of the software update system 10 according to the present embodiment. Fig. 2 is a system block diagram depicting the functional composing elements of the software update system 10 according to the present embodiment. The software update system 10 according to the present embodiment is comprised of a communication terminal 100, software management server 200 and software restore device 300 as shown in Fig. 1 and Fig. 2. Now the communication terminal 100, software management server 200 and software restore device 300 will be described in detail.

[0039]

The communication terminal 100 is physically comprised of a rewritable non-volatile memory 101, rewritable volatile memory 102, communication device 103, wire communication device 104 and CPU (Central Processing Unit) 105. For the communication terminal 100 according to the present embodiment, a wide variety of mobile communication terminal can be used, including a portable telephone, a portable terminal such as a PDA (Personal Digital Assistant), and a communication terminal such as a digital home electronic product which requires an update of

software, that is stored in the rewritable non-volatile memory 101 and is directly executed, by downloading the update file using the communication device 103.

[0040]

5 In the rewritable non-volatile memory 101, basic software for controlling the communication terminal 100, such as an OS, middleware, and communication software are stored, and this is a non-volatile memory where information stored in the memory can be rewritten like flash ROM, and
10 information stored in the memory is not lost even if the power is shut OFF. For the basic software stored in the rewritable non-volatile memory 101, software stored in the rewritable non-volatile memory 101 is executed in the rewritable non-volatile memory 101 without being developed into
15 another storage medium, unlike software stored in a hard disk, such as in a personal computer, which is read by another storage medium, such as memory, and is executed there. In other words, the basic software and the like, stored in the rewritable non-volatile memory 101 is directly executed in
20 the rewritable non-volatile memory 101.

[0041]

 The rewritable volatile memory 102 is a memory, for example RAM (Random Access Memory), which is used for storing data and software which do not affect operation of the
25 communication terminal 100 even if stored information is erased by a power OFF or another reason.

[0042]

The communication device 103 is a device for communicating with the software management server 200, and is used for transmitting identification information such as a terminal ID of the communication terminal 100 and the version information of the software, which is stored in the rewritable non-volatile memory 101 and is directly executed, of the communication terminal 100, to the software management server 200, or for receiving an update file from the software management server 200. The communication line which the communication device 103 is connected to may be either radio or such cable as a telephone line.

[0043]

The wire communication device 104 is a device for connecting with the software restore device 300 via such a cable as serial cable, and is used for receiving software from the software restore device 300 when software is transferred from the connected software restore device 300 to the rewritable non-volatile memory 101, and is restored.

[0044]

The CPU 105 is a central processing unit for controlling the communication terminal 100, and controls the communication terminal 100 according to the software stored in the rewritable non-volatile memory 101 or rewritable volatile memory 102.

[0045]

Now the functional composing elements of the communication terminal 100 will be described. As Fig. 2 shows, the communication terminal 100 is comprised of the functional composing elements of identification information storage section 111 (communication terminal identification information storage means, software identification information storage means), identification information transmission section 112 (communication terminal identification information transmission means, software identification information transmission means), update file information reception section 113 (update file information reception means), transfer status transition section 114 (transfer status transition means), transfer request transmission section 115 (transfer request transmission means), update file reception section 116 (update file reception means), software rewriting section 117 (software rewriting means), rewrite success judgment section 118 (rewrite success judgment means), restore status transition section 119 (restore status transition means), and restore software reception section 120 (restore software reception means). These composing elements are implemented by storing them in the rewritable non-volatile memory 101, for example, as software and data for controlling software updating, and the CPU 105 controls the communication device 103 and wire communication device 104 according to this software.

[0046]

The identification information storage section 111 stores the identification information, such as the terminal ID of the communication terminal 100 and the version information of software which is stored in the rewritable non-volatile memory 101 and is directly executed. Specifically the identification information storage section 111 is implemented by storing the identification information in the rewritable non-volatile memory 101 of the communication terminal 100.

[0047]

The identification information transmission section 112 acquires identification information, such as the terminal ID of the communication terminal 100 stored in the identification information storage section 111 and the version information of the software, which is stored in the rewritable non-volatile memory 101 and is directly executed, and transmits it to the software management server 200.

[0048]

The update file information reception section 113 receives update file information, including information on the presence of an update file and the size information of the update file, which is transferred from the software management server 200, from the software management server 200 via the communication device 103.

[0049]

The transfer status transition section 114 acquires the size information of the update file included in the update file information if an update file exists, based on the update file information received by the update file information reception section 113. And based on the size information of the update file, the transfer status transition section 114 limits operation of software which uses the rewritable volatile memory 102, and secures an area required for transferring and storing the update file in the rewritable volatile memory 102 so as to enable transfer of the update file.

[0050]

The transfer request transmission section 115 transmits notification to request transfer of an update file to the software management server 200, to the software management server 200 via the communication device 103 after the area required for storing the update file is secured in the rewritable volatile memory 102 by the transfer status transition section 114.

[0051]

The update file reception section 116 connects with the software management server 200 via the communication device 103 after the transfer request transmission section 115 transmits the above request to the software management server 200. And the update file reception section 116 receives the update file from the software management server 200 and stores it in the rewritable volatile memory 102.

[0052]

The software rewriting section 117 rewrites the software, which is stored in the rewritable non-volatile memory 101 and is directly executed, with the update software, which is stored in the rewritable volatile memory 102 as an update file, after the update file reception section 116 receives the update file from the software management server 200, and is stored in the rewritable volatile memory 102.

[0053]

The rewrite success judgment section 118 judges success of the processing of the software rewriting section 117 to rewrite the software, which is stored in the rewritable non-volatile memory 101 and is directly executed, with the update software which is stored in the rewritable volatile memory 102 as an update file.

[0054]

The restore status transition section 119 enables connection of the communication terminal 100 and the software restore device 300 via the wire communication device 104, when the rewrite success judgment section 118 judges that the rewriting of the software, executed by the software rewriting section 117, failed.

[0055]

The recovery software reception section 120 receives software from the software restore device 300 via the wire

communication device 104 when the restore status transition section 119 enabled connection of the software restore device 300 and the wire communication device 104 of the communication terminal 100. And the restore software reception section 120 stores the received software in the rewritable non-volatile memory 101, and restores the software of the rewritable non-volatile memory 101.

[0056]

Now the software management server 200 according to the present embodiment will be described. The software management server 200 according to the present embodiment is a computer system for managing update software to be stored in the rewritable non-volatile memory 101 of the communication terminal 100. The software management server 200 is physically comprised of a CPU (Central Processing Unit) 201, memory 202, hard disk 203, communication device 204, display device 205, and such an input device 206 as a keyboard and mouse, as shown in Fig. 1. Now out of the physical composing elements of the software management server 200, the communication device 204 will be described. The communication device 204 is a part for communicating with the communication terminal 100. Specifically, the communication device 204 is used for receiving the above identification information which is transmitted from the communication terminal 100, or transmitting the update file to the communication terminal

100. The communication line connected with the communication device 204 may be radio, or cable, such as a telephone line.

[0057]

5 The functional composing elements of the software management server 200 according to the present embodiment will now be described. The software management server 200 according to the present embodiment is comprised of the functional composing elements of the identification
10 information reception section 211, update software storage section 212, differential file creation section 213, update file information transmission section 214, transfer request reception section 215, and update file transmission section 216. These composing elements are configured as software
15 to be loaded into the memory 202 of the software management server 200, for example, and are implemented by the CPU 201 accessing the update software storage section 212 disposed on the hard disk 203, and controlling the communication device 205 according to the software. The functional composing
20 elements of the software management server 200 will now be described in detail.

[0058]

The identification information reception section 211 receives the above mentioned identification information,
25 which is transmitted by the identification information transmission section 112 of the communication terminal 100

via the communication device 204. The identification information reception section 211 outputs the received identification information to the differential file creation section 213.

5 [0059]

The update software storage section 212 stores the update software of the target communication terminal associating the terminal ID and version information of the software, and is disposed on the hard disk 203 of the software management server 200. Specifically, the update software storage section 212 stores the update software to be stored in the rewritable non-volatile memory 101 of the communication terminal and the file capacity thereof in association with the terminal ID of the communication terminal and version information of the update software, as shown in Fig. 3.

[0060]

The differential file creation section 213 creates a differential file between the update software to be stored in the rewritable non-volatile memory 101 of the communication terminal 100 which is stored in the update software storage section 212 and the software which is currently operated in the rewritable non-volatile memory 101 of the communication terminal 100 based on the terminal ID and version information of the software, which is stored in the rewritable non-volatile memory 101 and is directly executed, of the communication terminal 100, included in the identification

information which is output by the identification information reception section 211. Specifically, when the terminal ID of the communication terminal 100 included in the identification information is "0001", and the version number of the software which is currently operated in the rewritable non-volatile memory 101 of the communication terminal 100 is "1", the new update software stored in the update software storage section 212 has the latest version "2", and the differential file of versions "1" and "2" of the update software is created. The created differential file is output to the update file transmission section 216 as an update file, and update file information which includes the information notifying that an update file exists and the size information of the differential file is output to the update file information transmission section 214. In the present embodiment, the differential file is created, but if the band of the communication line of the software management system is wide enough to transfer a large sized file and the rewritable volatile memory 102 of the communication terminal has a capacity sufficient for storing the update software, even if a differential file is not created, then the update software with version "2" may be output to the update file transmission section 216, as is, as the update file, and 150 bytes, which is the size of the update software with version "2", may be included in the update file information as the size information, and is output to the update file information transmission section 214. On the

other hand, if the terminal ID of the communication terminal 100 included in the identification information is “0001”, and if there is no update software with a newer version, as in the case of when the version number of the software currently operated in the rewritable non-volatile memory 101 of the communication terminal 100 is “2”, then notice that the update software does not exist is included with the update file information, and is output to the update file information transmission section 214.

10 [0061]

The update file information transmission section 214 transmits the update file information received from the differential file creation section 213 to the communication terminal 100 via the communication device 204.

15 [0062]

The transfer request reception section 215 receives a request to transmit the update file, which is transmitted by the transfer request transmission section 115 of the communication terminal 100, via the communication device 204.

[0063]

The update file transmission section 216 transmits the update file to the communication terminal 100 via the communication device 204 based on the above mentioned request received by the transfer request reception section 215.

[0064]

Now the software restore device 300 according to the present embodiment will be described. The software restore device 300 according to the present embodiment is a dedicated device for restoring software by wire communication with the communication terminal 100 when the processing of rewriting the software of the rewritable non-volatile memory 101 of the communication terminal 100 failed. As shown in Fig. 1, the software restore device 300 is physically comprised of a CPU (Central Processing Unit) 301, memory 302, hard disk 303, wire communication device 304, display device 305, and such an input device 306 as a keyboard and mouse. Now out of the physical composing elements of the software restore device 300, the wire communication device 304 will be described.

[0065]

The wire communication device 304 is a part for performing communication by connecting the communication terminal 100 with a cable. Specifically, the cable connection device 304 is used for transmitting software to the communication terminal 100, and restoring the software in the rewritable non-volatile memory 101 of the communication terminal 100, or for acquiring identification information of the communication terminal 100 from the communication terminal 100.

[0066]

Now the functional composing elements of the

software restore device 300 according to the present embodiment will be described. As Fig. 2 shows, the software restore device 300 according to the present embodiment is comprised of the functional composing elements of an identification information acquisition section 311, restore software storage section 312, restore software selection section 313, and restore software transfer section 314. These composing elements are configured as software to be loaded into the memory 302 of the software restore device 300, for example, and is implemented by the CPU 301 accessing the restore software storage section 314 which is disposed on the hard disk 303, or controlling the wire communication device 304. Now the functional composing elements of the software restore device 300 will be described.

[0067]

The identification information acquisition section 311 acquires the identification information of the communication terminal 100 by connecting with the wire communication device 104 of the communication terminal 100 via the wire communication device 304.

[0068]

The restore software storage section 312 stores software in a similar format as the update software storage section 212 of the software management server 200, shown in Fig. 3. The restore software storage section 312 is disposed on the hard disk 303 of the software restore device 300.

[0069]

5 The restore software selection section 313 selects software to be transferred to the rewritable non-volatile memory 101 of the communication terminal 100 for restoration, from the restore software storage section 312 based on the identification information of the communication terminal 100 acquired by the identification information acquisition section 311.

[0070]

10 The restore software transmission section 314 transmits the software selected by the restore software selection section 313 to the communication terminal 100 via the wire communication device 304.

[0071]

15 Now operation of the software update system according to an embodiment of the present invention will be described, and also the update method for software of the software update system according to the present embodiment will be described. Fig. 4 is a flow chart depicting the update
20 method for software according to the present embodiment.

[0072]

25 The identification information transmission section 112 of the communication terminal 100 transmits the identification information to the software management server 200 (step S01). The identification information reception section 211 of the software management server 200 receives

the identification information transmitted by the identification information transmission section 112 (step S02), and outputs it to the differential file creation section 213. Based on the terminal ID of the communication terminal 100 and version information of the software included in the identification information, the differential file creation section 213 creates the differential file between the update software and the software which is stored in the rewritable non-volatile memory 101 of the communication terminal 100 and is directly executed, if there is an update software to be transmitted from the update software storage section 212 to the communication terminal 100 (step S03), and outputs this differential file to the update file transmission section 216 as an update file. The differential file creation section 213 also outputs update file information which includes information to notify that an update file exists and the size information of this update file to the update file information transmission section 214. If an update file does not exist, on the other hand, the differential file creation section 213 outputs the update file information, to notify that an update file does not exist, to the update file information transmission section 214. As mentioned above, if the band of the communication line of the software management system is wide enough to transfer a large sized file, and if the rewritable volatile memory 102 of the communication terminal has a capacity sufficient for storing the update software, even if a differential file is not

created, then the update software can be transmitted, as is, as an update file.

[0073]

Then the update file information transmission section
5 214 transmits the update file information received from the
differential file creation section 213 to the communication
terminal 100 (step S04). The update file information
reception section 113 of the communication terminal 100
10 receives the update file information transmitted by the update
file information transmission section 214, and judges whether
an update file exists (step S05). If an update files does not
exist, the update file information reception section 113
disconnects communication between the communication
terminal 100 and the software management server 200 (step
15 S06), and ends software update processing. If an update file
exists, on the other hand, the transfer status transition section
114 limits operation of software which uses the rewritable
volatile memory 102 of the communication terminal 100
based on the size information included in the update file
20 information, and secures an area required for transferring the
update file so as to enable transfer of the update file (step
S07). After the status shifts to the update file transfer
enabled status by the transfer status transition section 114, the
transfer request transmission section 115 transmits a request
25 to transmit the update file to the software management server
200 (step S08). The transfer request reception section 215 of

the software management server 200 receives the request transmitted from the transfer request transmission section 115 (step S09), and based on this request, the update file transmission section 216 transmits the update file to the communication terminal 100 (step S10). The update file reception section 116 of the communication terminal 100 receives the update file transmitted by the update file transmission section 216 of the software management server 200, and stores it in the rewritable volatile memory 102 (step S11). When storing the update file in the rewritable volatile memory 102 by the update file reception section 116 completes, communication between the communication terminal 100 and the software management server 200 is disconnected (step S12).

[0074]

Then the software rewriting section 117 rewrites the software, which is stored in the rewritable non-volatile memory 101 and is directly executed, with the update software which is stored in the rewritable volatile memory 102 as an update file (step S13). The rewrite success judgment section 118 judges whether rewriting of the software by the software rewriting section 117 succeeded (step S14), and if rewriting succeeded, the rewrite success judgment section 118 ends the software update processing.

[0075]

If the rewrite success judgment section 118 judges that

the update of software by the software rewriting section 117 failed, on the other hand, the restore status transition section 119 enables connection of the communication terminal 100 and the software restore device 300 via the wire communication device 104 of the communication terminal 100 and the wire communication device 304 of the software restore device 300 (step S15).

[0076]

When the communication terminal 100 and the software restore device 300 are connected via a cable to restore the communication terminal 100 which operation is disabled due to a failure of update of the software, the identification information acquisition section 311 of the software restore device 300 acquires the identification information of the communication terminal 100 from the identification information storage section 111 of the communication terminal 100 (step S16). Based on the identification information acquired by the identification information acquisition section 311, the restore software selection section 313 selects the software to be transferred to the rewritable non-volatile memory 101 of the communication terminal 100 from the restore software storage section 312 (step S17). The restore software transmission section 314 transmits the software selected by the restore software selection section 313 via the wire communication device 304 (step S18). The restore software reception section 120 of the

communication terminal 100 receives the software transmitted
by the restore software transmission section 314 (step S19),
and stores this software in the rewritable non-volatile memory
101, and the software, which is stored in the rewritable
5 non-volatile memory 101 and is directly executed, is restored
(step S20).

[0077]

The functions and effects of the software update
system according to an embodiment of the present invention
10 will now be described.

[0078]

After the update file is transferred from the software
management server 200 to the rewritable volatile memory 102,
the software, which is stored in the rewritable non-volatile
15 memory 101 and is directly executed, is rewritten with the
update software which is stored as an update file. Therefore
the rewritable non-volatile memory need not be duplicated for
updating the software. As a result, the communication
terminal 100 can be downsized and cost can be decreased.
20 Also transfer of an update file which tends to take time and
have such a risk factor as power disconnection, and rewriting
of software which takes shorter time than above can be
separated, so the risk of failing to update software can be
decreased. Also even if transfer is interrupted in the middle
25 of transferring an update file, the software, which is stored in
the rewritable non-volatile memory 101 of the communication

terminal 100 and is directly executed, has not been updated, so operation of the communication terminal 100 is not affected.

[0079]

5 Also operation of the software which uses the rewritable volatile memory 102 is limited in the normal operating status of the communication terminal 100 before an update file is transferred, then the update file is transferred to the rewritable volatile memory 102. Therefore even if the
10 capacity of the rewritable volatile memory 102 is small, the update file can be transferred to the rewritable volatile memory 102.

[0080]

15 Also the area for storing the update file is secured in the rewritable volatile memory 102 based on the size information of the update file included in the update file information. Therefore an appropriate size of the area for storing the update file can be secured in the rewritable volatile memory 102.

[0081]

20 Also the communication terminal 100 transmits the identification information of the software, which is stored in the rewritable non-volatile memory 101 and is directly executed, and the identification information of the
25 communication terminal 100 to the software management server 200. Therefore the software management server 200,

which received this identification information, can specify the model of the communication terminal 100 based on the identification information of the communication terminal 100, and can specify the version of the software which is currently
5 operated in the rewritable non-volatile memory 101 of the communication terminal 100. As a result, out of the update software owned by the software management server 200, the update software to be stored in the rewritable non-volatile memory 101 of the communication terminal 100 can be
10 accurately selected. Also the software management server 200 creates a differential file between the software, which is stored in the rewritable non-volatile memory of the communication terminal 100 and is directly executed, and the update software owned by the software management server
15 200, based on the identification information of the software, which is stored in the rewritable non-volatile memory 101 and is directly executed, and uses this differential file as an update file. Therefore the size of the update file to be transmitted can be decreased. As a result, the update file can
20 be accurately transmitted even via a narrowband communication line.

[0082]

Since the present invention has the above mentioned functions, the update software can be efficiently transferred
25 even via a narrowband communication line, such as a radio communication line. Also even if it takes time to transfer an

update file, this does not become a cause of failing to update software, therefore the risk of failing to update software is low. As a consequence, the rewritable non-volatile memory of the communication terminal 100 need not be duplicated for updating software.

[0083]

Also even if rewriting of software failed due to a power failure, for example, which occurred in the middle of the processing of rewriting the software, which is stored in the rewritable non-volatile memory 101 and is directly executed, with the update software, wire communication with the software restore device 300 for restoring the software, which is stored in the rewritable non-volatile memory 101 and is directly executed, is enabled by the restore status transition means 120, so if the communication terminal 100 and the software restore device 300 are connected, software can be restored in the rewritable non-volatile memory 101 by the software restore device 300.

[0084]

[Effects of the Invention]

According to the present invention, the update file is transferred to the rewritable volatile memory of the communication terminal from the software management server, and then the software, which is stored in the rewritable non-volatile memory of the communication terminal and is directly executed, is rewritten with the update software which

is stored in the rewritable volatile memory as an update file. Therefore transfer of an update file, which takes time and tends to have such a risk factor as power disconnection and the rewriting of software which takes a shorter time than
5 above, can be separated, and the risk of failing to update software can be decreased. Also the rewritable non-volatile memory of the communication terminal need not be duplicated to update software. As a result, decreasing the risk of failing to update software, which is the object of the
10 present invention, can be implemented, and also the object of downsizing and decreasing cost of the communication terminal can be achieved.

[Brief Description of the Drawings]

[Fig. 1]

15 Fig. 1 is a system block diagram depicting the physical composing elements of the software update system according to an embodiment of the present invention.

[Fig. 2]

20 Fig. 2 is a system block diagram depicting the functional composing elements of the software update system according to an embodiment of the present invention.

[Fig. 3]

25 Fig. 3 is a diagram depicting the storage format of the update file in the update software storage section of the software management server according to an embodiment of the present invention.

[Fig. 4]

Fig. 4 is a flow chart depicting the software update method by the software update system according to an embodiment of the present invention.

5 [Explanation of Reference Numerals]

10: software update system

100: communication terminal

101: rewritable non-volatile memory

102: rewritable volatile memory

10 103: communication device

104: wire communication device

105: CPU

111: identification information storage section

112: identification information transmission section

15 113: update file information reception section

114: transfer status transition section

115: transfer request transmission section

116: update file reception section

117: software rewriting section

20 118: rewrite success judgment section

119: restore status transition section

120: restore software reception section

200: software management server

201: CPU

25 202: memory

203: hard disk

	204: communication device
	205: display device
	206: input device
	211: identification information reception section
5	212: update software storage section
	213: differential file creation section
	214: update file information transmission section
	215: transfer request reception section
	216: update file transmission section
10	300: software restore device
	301: CPU
	302: memory
	303: hard disk
	304: wire communication device
15	305: display device
	306: input device
	311: identification information acquisition section
	312: restore software storage section
	313: restore software selection section
20	314: restore software transfer section

[Document Name] ABSTRACT

[Abstract]

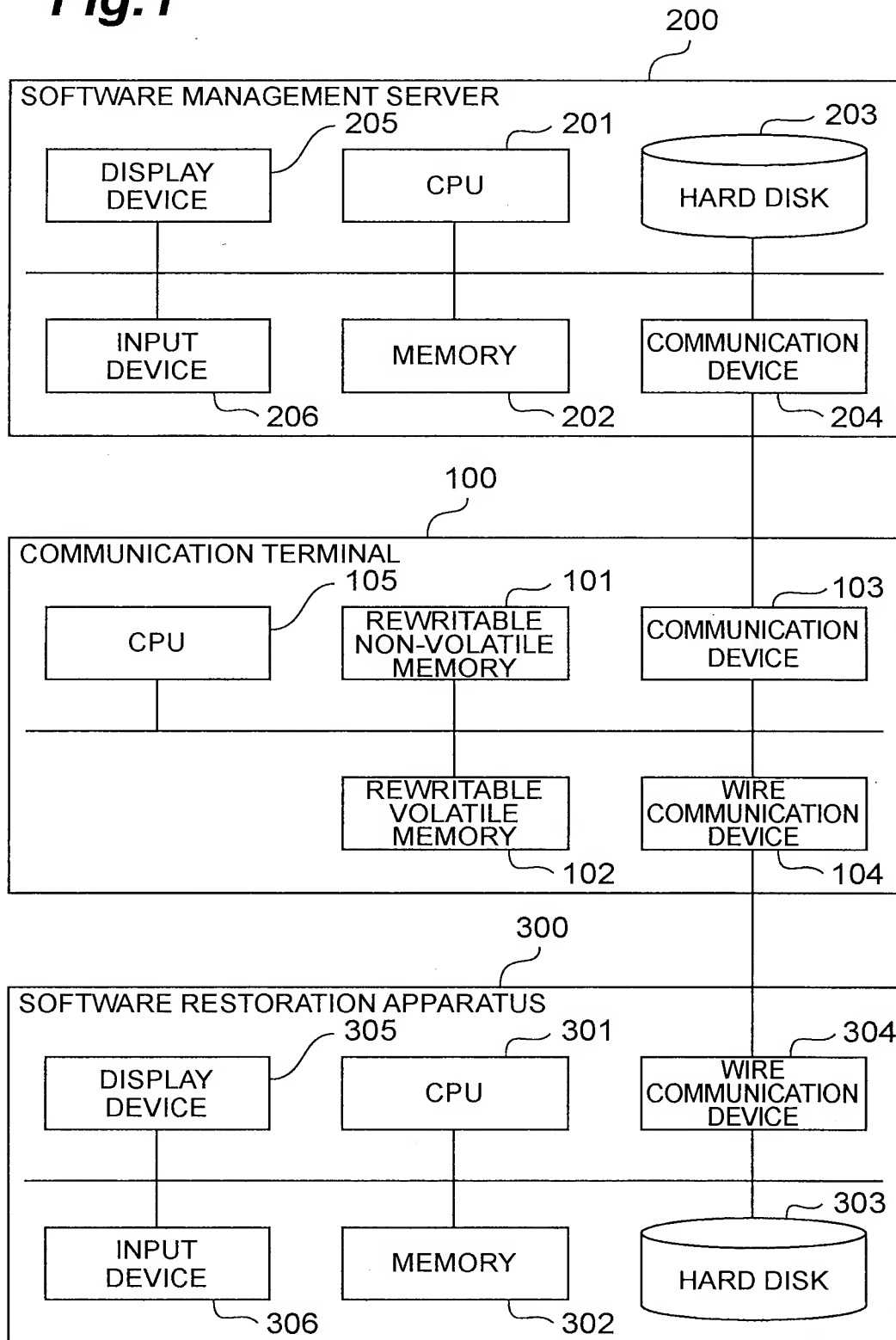
[Problem] The present invention provides an update method for software which has the low risk of failing to update software, even if a rewritable non-volatile memory of the communication terminal is not duplicated.

[Means of Solution] The present invention is an update method for software, which is stored in a rewritable non-volatile memory of a communication terminal and is directly executed, comprising an update file transfer step of transferring an update software from a software management server for managing the update software to be stored in the rewritable non-volatile memory to the communication terminal as an update file, and storing the update file in the rewritable volatile memory of the communication terminal, and a software rewriting step of rewriting the software which is stored in the rewritable non-volatile memory and is directly executed with the update software which is stored in the rewritable volatile memory as the update file after the update file transfer step completes.

[Selected Drawing] Fig.4

Fig.1

10



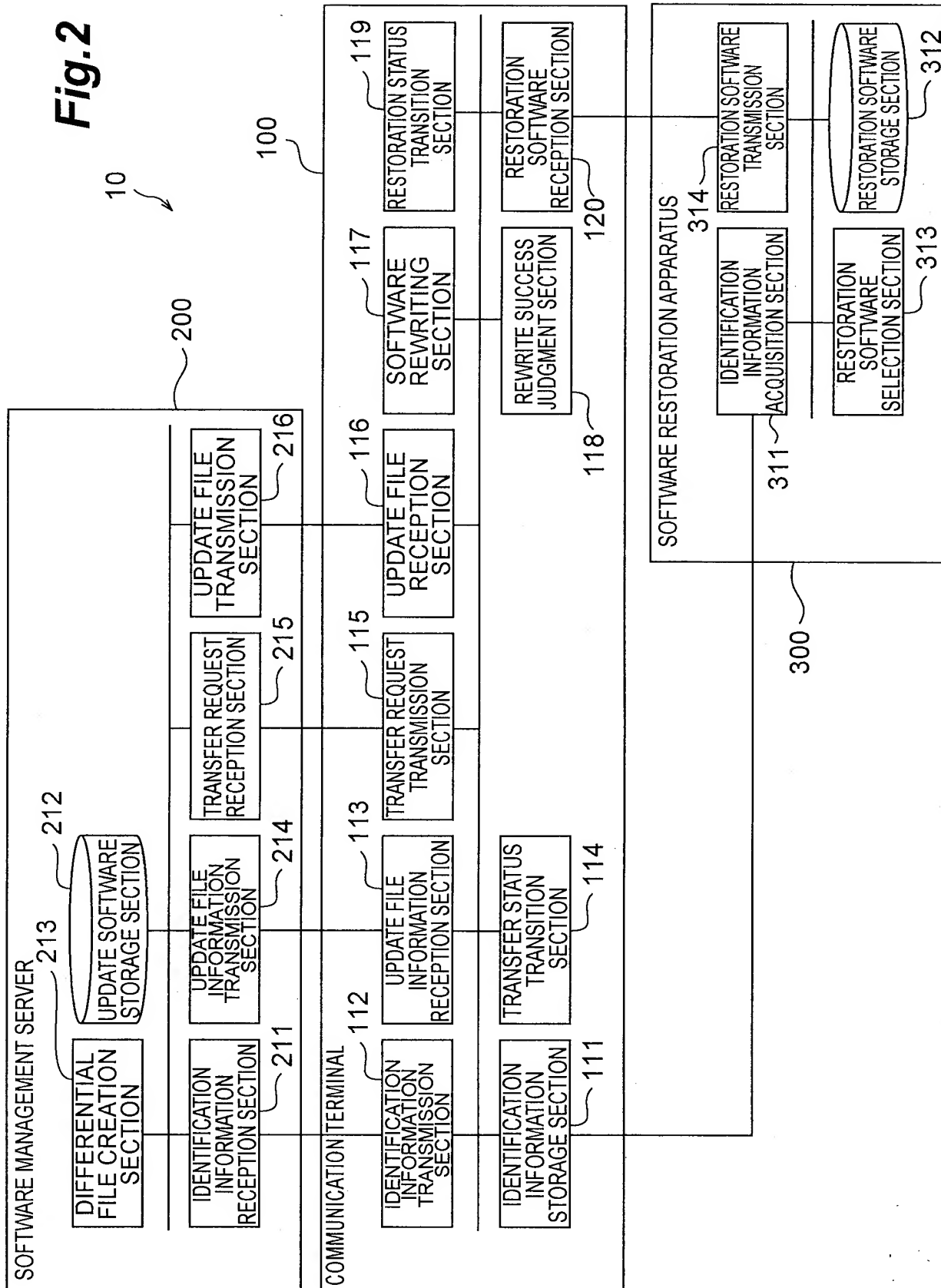


Fig.3

TERMINAL ID	VERSION INFORMATION	UPDATE SOFTWARE	FILE CAPACITY
0001	1	A	200
	2	B	150
0002	1	C	120
	2	D	180
⋮	⋮	⋮	⋮

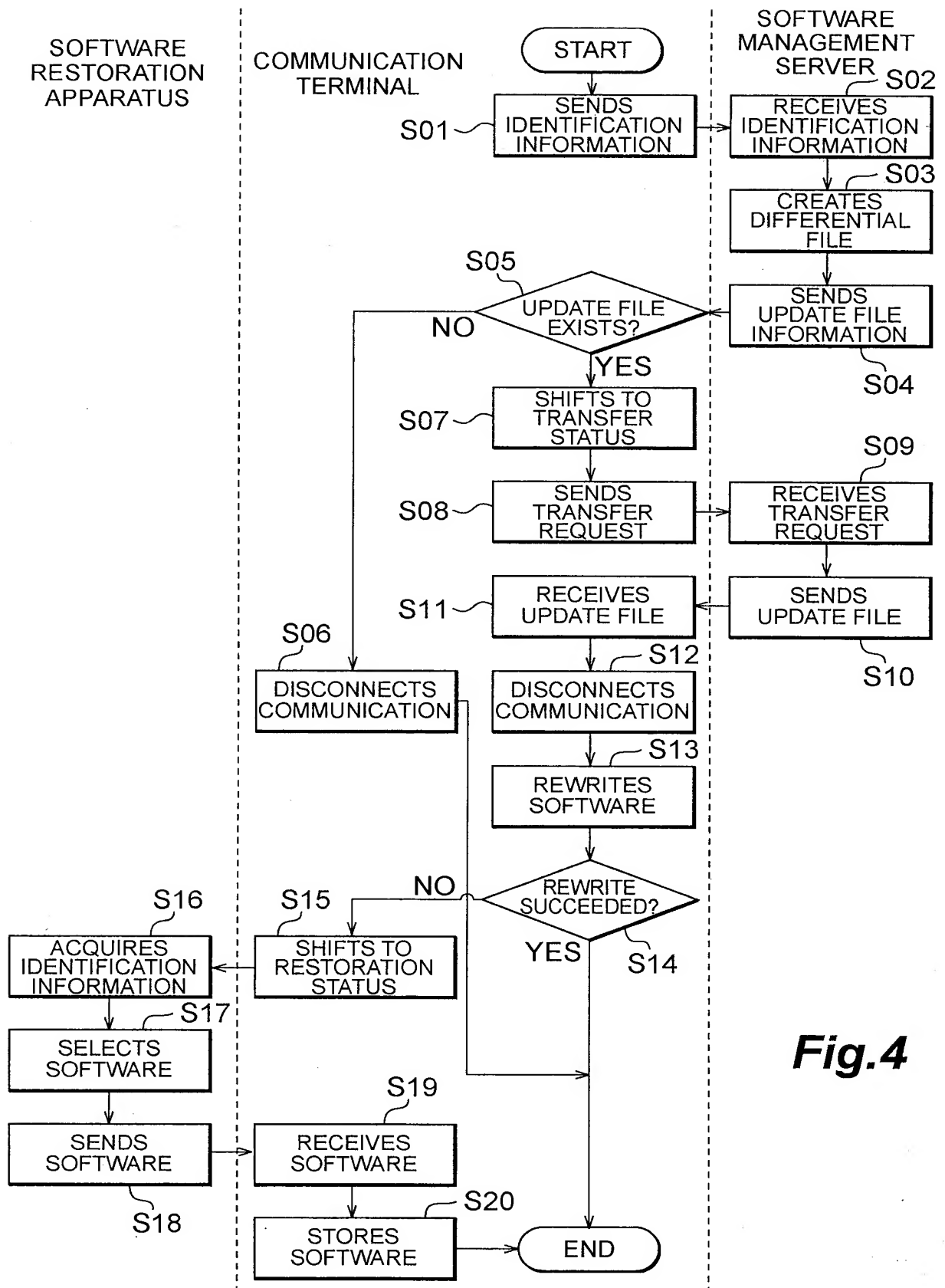


Fig.4